



Big Data Projekte

Erfahrungsbericht aus der echten Welt

Nico Kreiling

Karlsruhe, 15.9.2016



Nico Kreiling

Big Data Scientist

- › Wirtschaftsingenieurswesen am KIT,
Abschluss März 2016
- › Erfahrungen aus Web-Development und
Datenbankadministration
- › Bei der Inovex seit Juni 2016

Agenda

- › Ein Big Data Projekt
- › Technologiestack
- › Erfahrungen aus der Praxis

Ausgangssituation

- › Drei verschiedene Datenkategorien aus dem Webtracking: Clicks, Views und Events
- › Tägliches Datenaufkommen > 1 Mrd
- › Regelbewirtschaftung durch ETL-Strecken
- › Schaffen einer Analyseplattform für Data Scientists

Big Data?

Was sind viele Daten Big-Data?

Volume

- 6 TB

Velocity

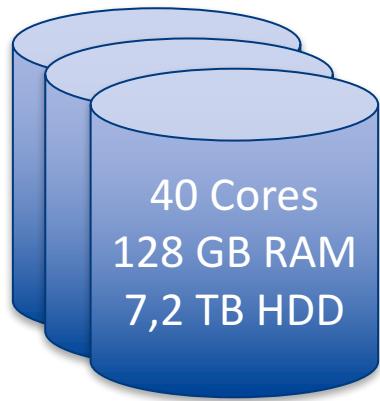
- ca. 1 Mrd/d
- Batch-Verfahren (h)

Variety

- ca. 100 Attribute
- Standardtypen

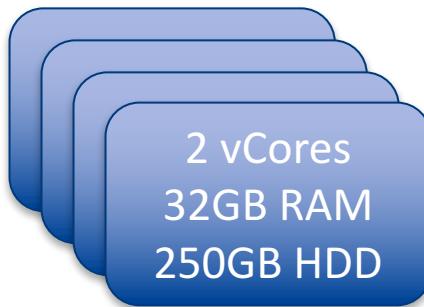
Big Data?

Mehr als auf einen Server passt



40 Cores
128 GB RAM
7,2 TB HDD

3 Server



2 vCores
32GB RAM
250GB HDD

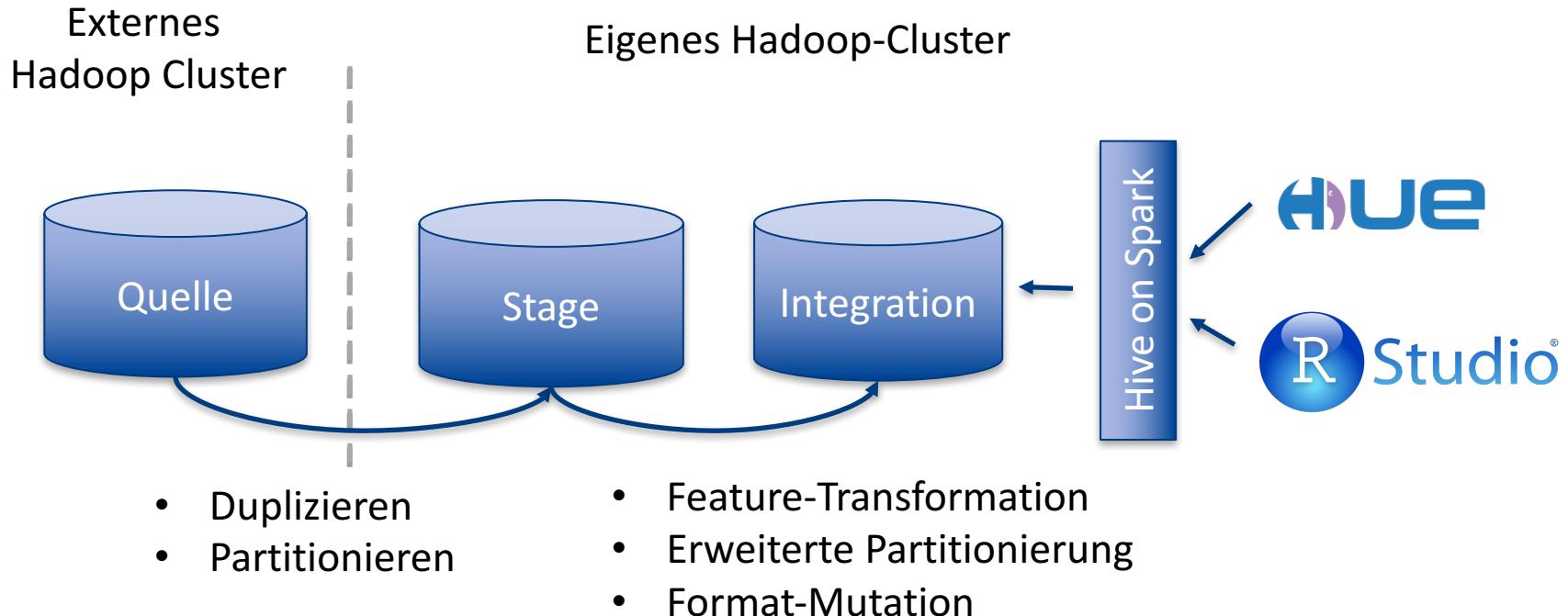
4 VMs Operativ



2 vCores
32GB RAM
300GB HDD

2 VMs Nutzer

Beispiel einer "Big Data"-Architektur



Technologie-Stack: Das Fundament

MapReduce

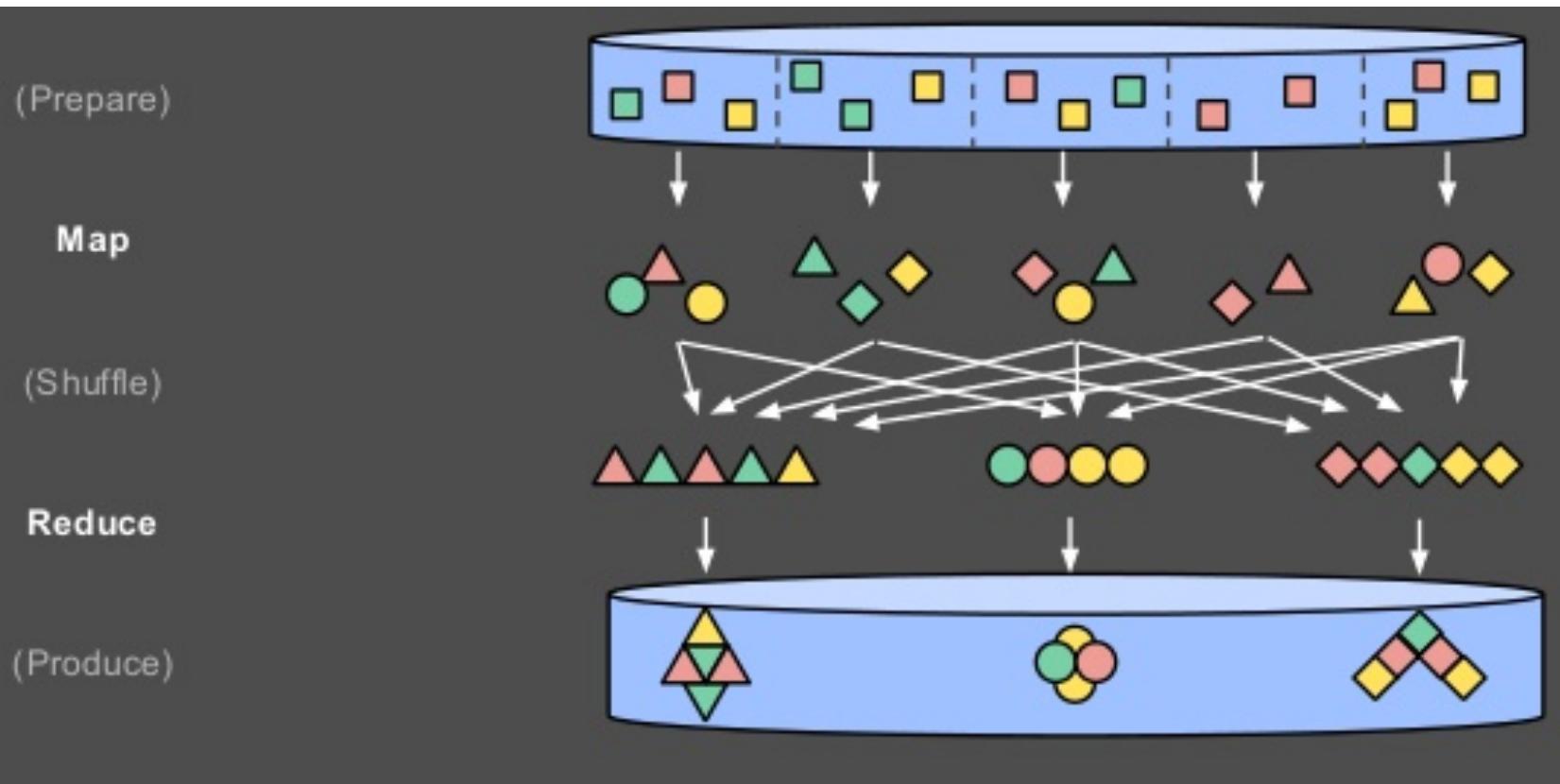
HDFS

Hadoop Ökosystem

HDFS: Verteiltes Dateisystem, welches Dateiblöcke einer fixen Größe redundant speichert.

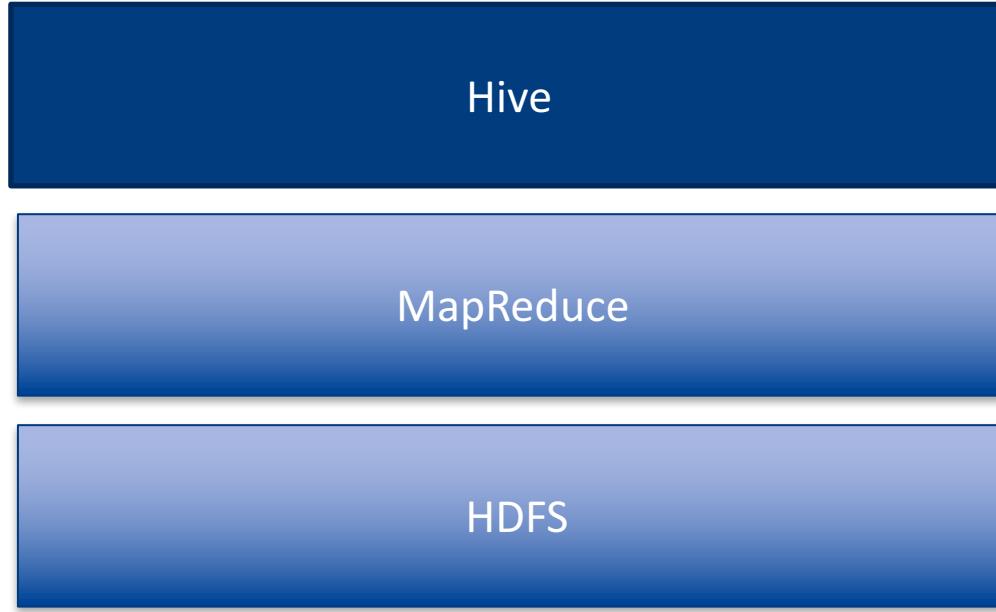
Map-Reduce: Parallelisierungsverfahren für verteilte Berechnungen

Map-Reduce Funktionsprinzip



Technologie-Stack: Datenaggregation

MySQL (Meta-Storage)



Hive

Hive: MapReduce-Generator basierend auf SQL Syntax

MySQL: Relationale Datenbank, wird hier als Metadaten-Speicher genutzt

Abfragsyntax: Hive vs MapReduce

```
package org.myorg;
import java.io.IOException;
import java.util.*;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {
    public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        public void map(LongWritable key, Text value, Context context)
            throws IOException, InterruptedException {
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                word.set(tokenizer.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {
        public void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            context.write(key, new IntWritable(sum));
        }
    }

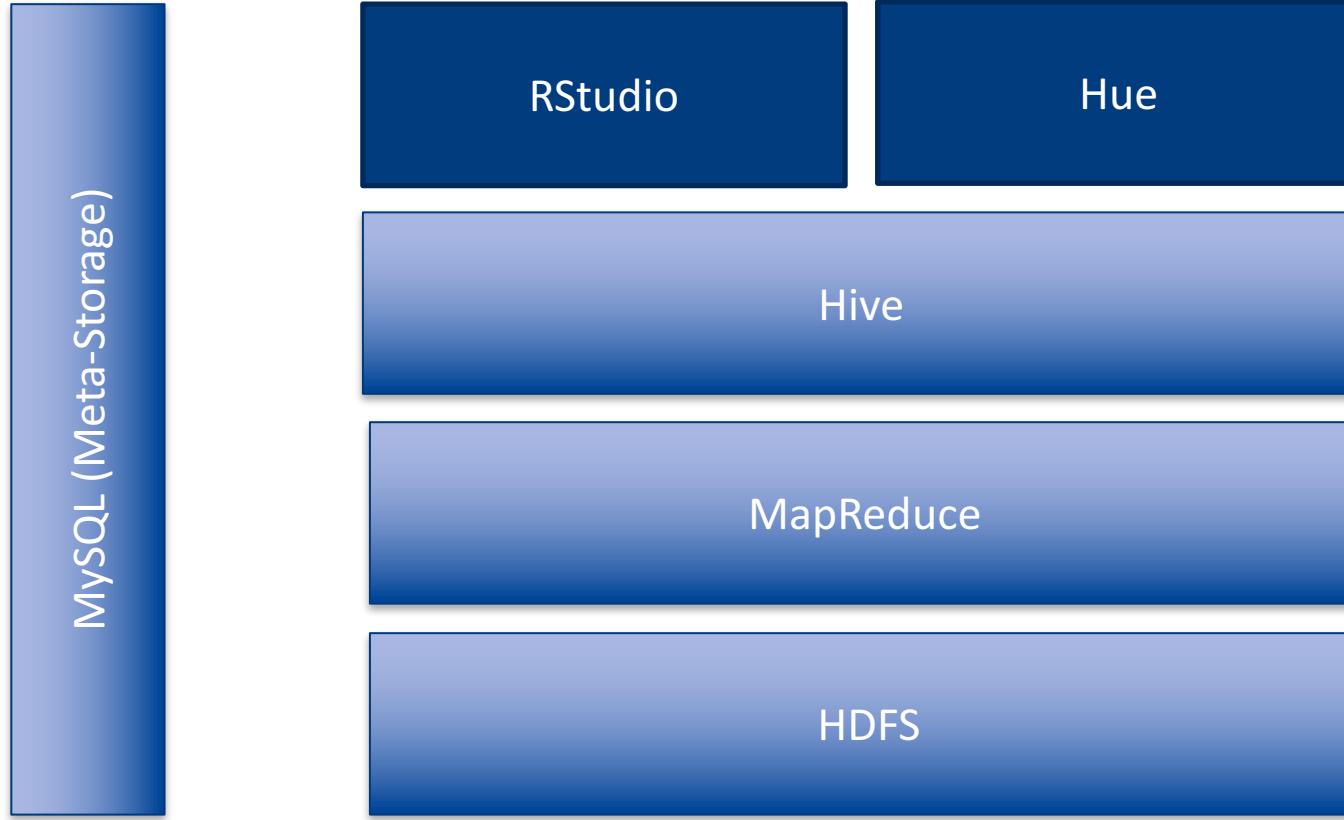
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = new Job(conf, "wordcount");
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        job.waitForCompletion(true);
    }
}
```

```
CREATE TABLE docs (line STRING);

LOAD DATA INPATH 'docs' OVERWRITE INTO TABLE docs;

CREATE TABLE word_counts AS
SELECT word, count(1) AS count
FROM (
    SELECT explode(split(line, '\s')) AS word
    FROM docs) w
GROUP BY word
ORDER BY word;
```

Technologie-Stack: Auswertung



Auswertungstools: Hue & RStudio

The image displays two data analysis environments side-by-side. On the left is the Hue interface, featuring a top navigation bar with 'Hive', 'Metastore Manager', and 'Search' tabs, and a main area for 'Hive Editor' queries. The central part of the screen shows a query editor with a code snippet for selecting events from a specific date range and hash contract ID. Below the editor are sections for 'SETTINGS', 'FILE RESOURCES', 'UDFS', and 'OPTIONS'. A large, stylized 'HUE' logo is partially visible at the bottom left. On the right is the RStudio interface, which includes a 'Console' window showing the R version (2.14.1) and its license terms, and a 'Workspace' window showing a file browser with a folder named '.Rhistory' and other media files like 'Downloads', 'My Music', 'My Pictures', and 'My Videos'. The RStudio logo is prominently displayed in large blue letters.

select event_ts_utc, event_rough, event_medium, event_fine, event_system, event_source_connection, event_country_code
from all_events
where hash_contract_id = "144f980250c0dd6537daa983fd884220"
and hash_account_id = "2d3307c6df5e1fcf509977b818c98629"

1 2016-01-04 07:41:15.0 Call_In Call_In|HANDLED Call_In|HANDLED|DE_iu1_WH_FO_1st CDR
2 2016-01-04 07:50:22.0 Call_In Call_In|HANDLED Call_In|HANDLED|DE_iu1_OHNL_1st CDR
3 2016-01-04 08:21:54.0 Call_In Call_In|HANDLED Call_In|HANDLED|DE_iu1_WH_FO_1st CDR
4 2016-01-04 08:32:34.0 Call_In Call_In|HANDLED Call_In|HANDLED|DE_iu1_WH_FO_1st CDR
5 2016-01-04 08:51:59.0 Call_In Call_In|HANDLED Call_In|HANDLED|DE_iu1_WH_FO_1st CDR
6 2016-01-04 09:00:10.0 Call_In Call_In|HANDLED Call_In|HANDLED|DE_iu1_WH_FO_1st CDR
7 2016-01-04 09:05:50.0 Call_In Call_In|HANDLED Call_In|HANDLED|DE_iu1_WH_FO_1st CDR

R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain
conditions.
Type 'license()' or 'licence()' for distribution details.

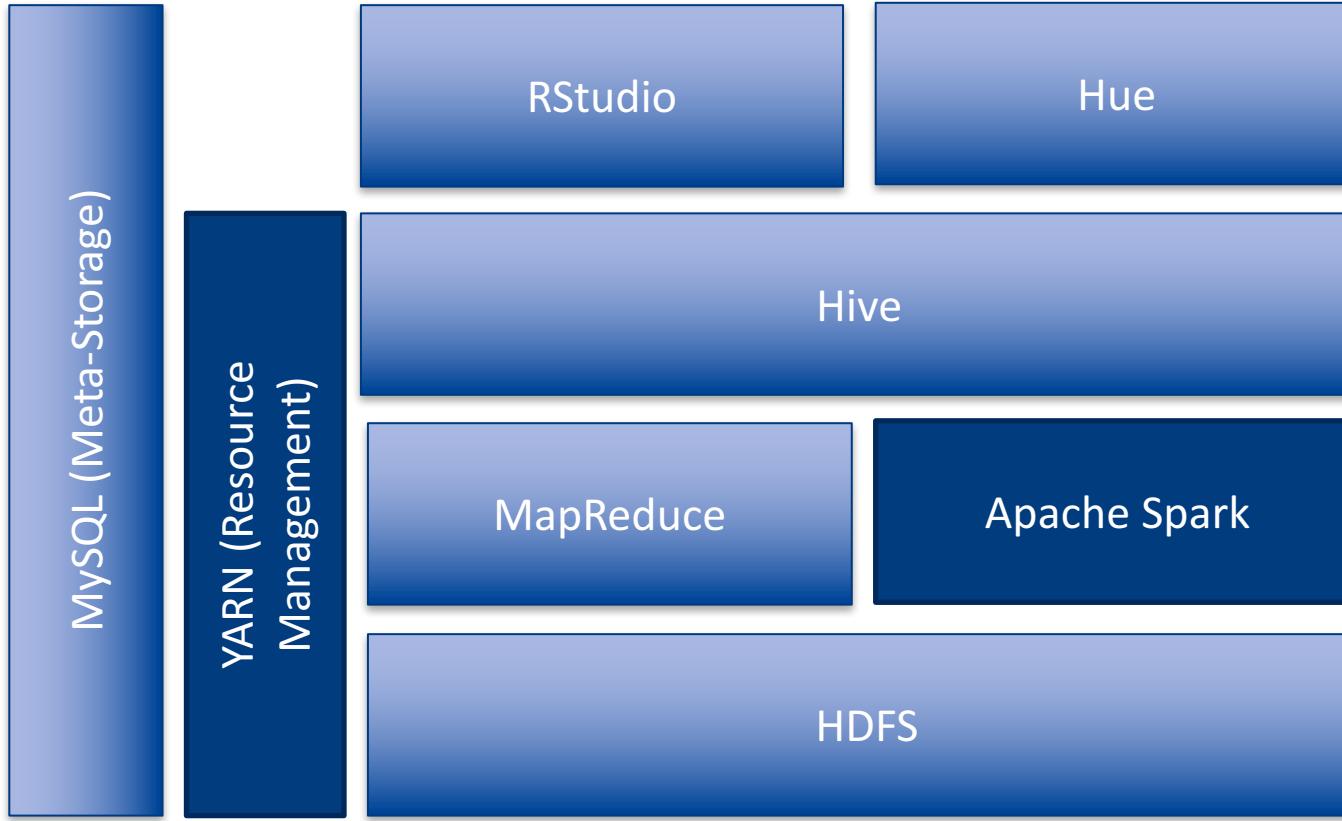
Natural language support but running in an English
locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in
publications.

Type 'demo()' for some demos, 'help()' for on-line help,
or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>

Technologie-Stack

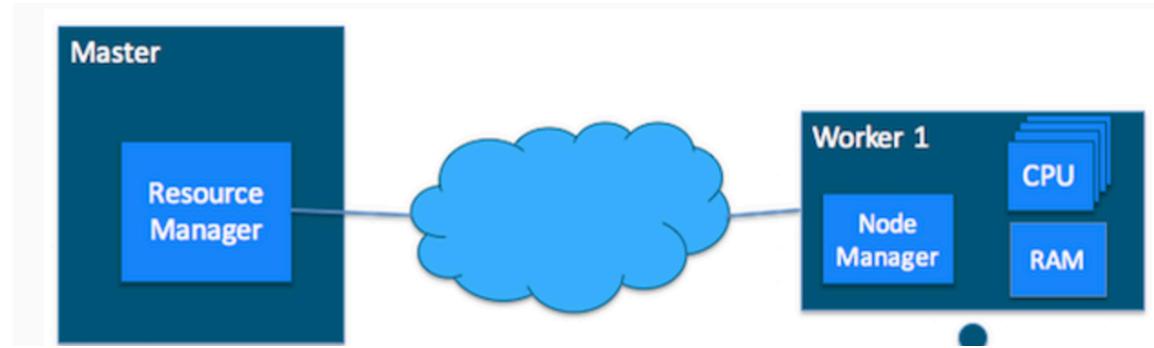


YARN

Yet Another Resource Negotiator

ResourceManager: Manages Resources

NodeManager: Handles the process execution

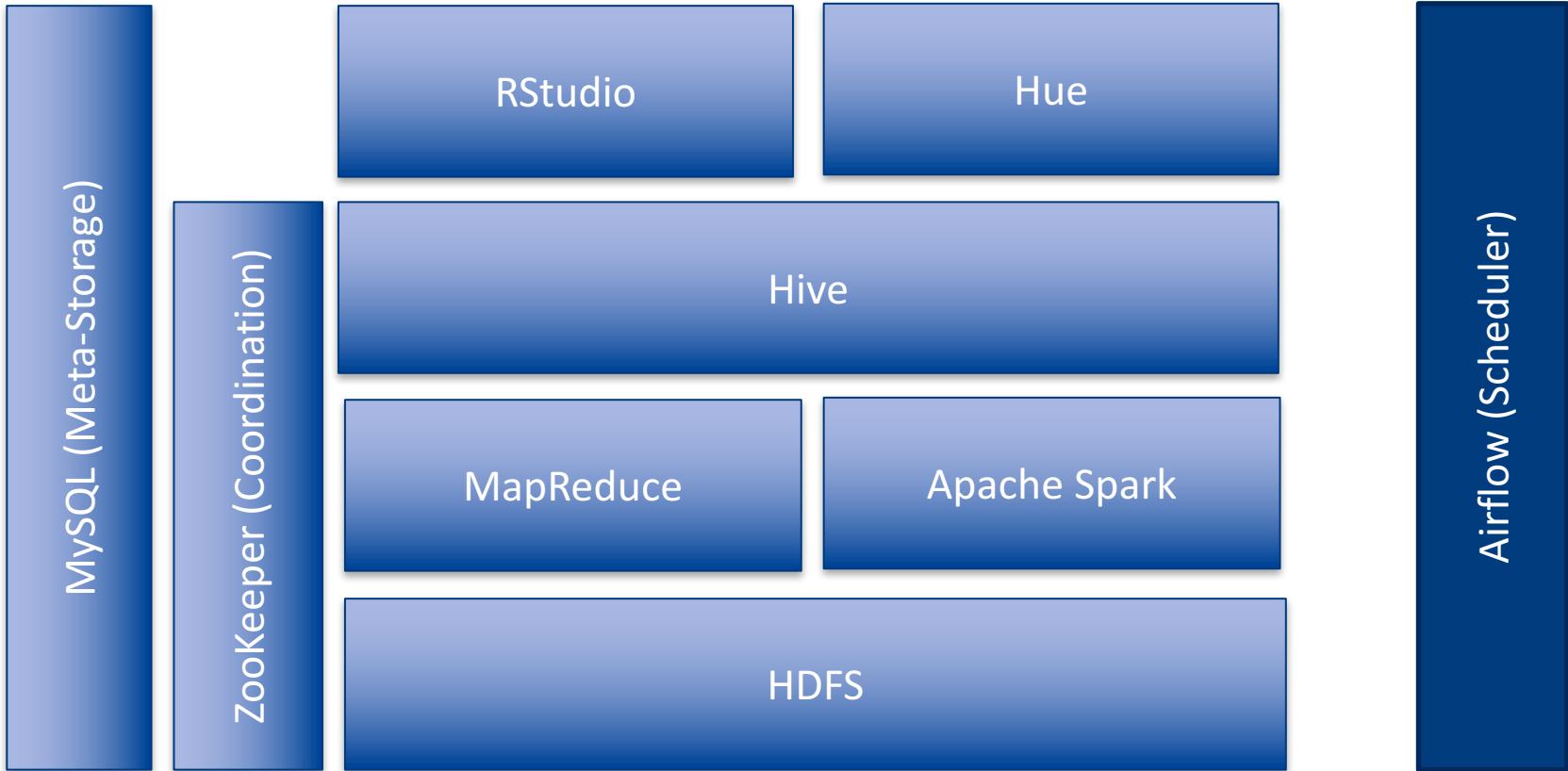


Spark



- In-Memory Berechnungen
- Graphstruktur (RDD)
- Lazy Evaluation

Technologie-Stack



Airflow

Workflow- und Job-Steuerung
Open Sourced von Airbnb



Praxiserfahrungen

- › Initiales Setup ist ein wesentlicher Aufwand
- › Low-Level IT-Wissen wichtig (shell, bash, OS-Befehle)

Echte Welt ≠ Hello World

Praxiserfahrungen – Lektion 1

- › Mehr Daten > mehr Ressourcen > weniger Übersicht
- › Struktur organisieren (Link-Page)
- › TMUX

Diverses Technologiespektrum

Praxiserfahrungen – Lektion 2

Datenbanken:

- MySQL
- Sqlite
- Oracle

Sprachen:

- Python
- Shell
- SQL
- YAML

Hadoop:

- Map-Reduce
- HDFS
- Hive
- Yarn

Sonstiges

- Airflow
- Ansible
- Hue

Noch diverse Auswahlmöglichkeiten

Praxiserfahrungen

MySQL PostgreSQL Oracle CouchDB MongoDB
CouchDB Neo4j Elasticsearch Solr Redis Memcached
Riak Dynamo Swoop Accumulo Cassandra HBase Yarn
Kubernetes Mesos Tensorflow TitanDB DockerSwarm
Flink Storm Apache Streams Tez Spark MapReduce
Impala Hive Pig Drill MicroStrategy Tableau Caravel
Maven Pip Apt Airflow kaleidoscope Pandas Kerberos
Kafka Flume Kibana NumPy Rstudio NiFi Solr Zeppelin
Gearpump Greenplum

Lernen effizient gestalten

Praxiserfahrungen – Lektion 2

Linux-Utilities & Shell

Shell FreqUsed

```
```sh
less $(locate my.cnf)
perl -pi -e 's/\r//' ~/.tmux.conf # Change
Windows line-endings to Linux
type -a sqlite # Show destination of alias
```
```

OS-Information

```
```sh
cat /etc/*-release # Show Linux Version
printenv # print all System variables
apt list --installed # Show installed packages
```
```

Linux-Utilities & Shell

Shell FreqUsed

```
less $(locate my.cnf)
perl -pi -e 's/\r//' ~/.tmux.conf # Change
type -a sqlite # Show destination of alias
```

OS-Information

```
cat /etc/*-release # Show Linux Version
printenv # print all System variables
apt list --installed # Show installed packages
```

Testaufwand minimieren

Praxiserfahrungen – Lektion 3

- › Wiederkehrende Kommandos in Skripte auslagern
- › Kürzel in .bashrc definieren
- › KeePass AutoType für Passwörter nutzen

Fazit

- › Big-Data Projekte gibt es wirklich!
- › Schnelle technische Weiterentwicklung und diverses Portfolio an Frameworks
- › Erforderliches Skillset recht divers
- › Viele Möglichkeiten sich zu verwirklichen

Vielen Dank

Nico Kreiling

Big Data Scientist

inovex GmbH

Ludwig-Erhard-Allee 6
76131 Karlsruhe

nico.kreiling@inovex.de

0173 3181 166